Vision

Designing and implementing a simulation for customers to simulate course registration system. With the help of the simulation, customers can find out how the course registration process works and observe the problems that may occur in during process.

Scope

- This simulation provides statistical data about courses and students in the end of the registration process.
- This simulation provides information about specific problems during the registration process.

Iteration Plan

First Iteration: In the first iteration, we will spend our energy and focus into documentation phase mostly. We will hold several meetings to agree on core functionalities on the general concept of the project. After, we will assign small groups for the analysis and design phase. We will start preparing the requirement analysis document. After that, we will work on design phase. When we think that the documentation is sufficient for starting the coding phase, we will start coding according to the analysis and design phase implications.

Glossary of Term

Student: Main actor of the system.

Transcript: Detailed and approved record of students grades that has been generated by university.

Registration Process: Advisor-controlled course selection process of student's enrollment.

Curriculum: The subjects comprising a course of study in a university.

Semester: A half year term in a university.

Prerequisite Tree: It is the model that contains the pre-conditional relations between the courses in the curriculum. In this project we will simulate prerequisite tree for the students who registered in 2020 and later.

Advisor: University academic member who is approving the students' academic program.

Quota: A limited number of students that can enroll to specific course.

Course: Syllabus item offered by the university.

Credit: The term that measures the numerical value of a course based on hours in a week or the content of the course, used for selecting some of the elective courses and graduation.

Functional Requirements

- Certain number of students for each semester should be generated randomly with the given fields:

    Student ID, GPA, Name, Surname and Transcript

- Registration process should be simulated for each student that are generated randomly.
- Problems that may occur regarding course registration should be checked.
- Successfully registered courses should be recorded on the student's transcript.
- If a student fails to register a course the reason behind it should be written to its json file.
    - to register a course, student must pass its pre-requisites.
    - to register a course, that course's quota must not be exceeded.
    - Student cannot enroll courses from an upper semester.

- An output file containing general statistics about the course registration process should be created on a departmental basis

Non-Functional Requirements

- Simulation should be implemented using Java programming language.
- Considering the addition of different iterations, the program has been developed in accordance with the iterative progress principles.
- The program will use json format both inputs and outputs files.
- The simulation will not be supported by user interface. All outputs will be accessed from command-line and json files.
- Outputs and logs of the system must be proper manner and to be understandable clearly.

Use-case

Scenario: Registration to Course

Actors: Student, Advisor

1- All parameters required for the system to work will be included in the input file such as courses, semesters, and student names.
2- The system will parse the input file.
3- The system will generate random students with equal probability in any of the 8 semesters.
4- The system will assign each student an advisor randomly.
5- The system will assign a successful/failed course, considering the prerequisites.
6- The system will generate a transcript for each student and store course information it in output file.
7- The system will then start the registration process for the relevant semester.
8- The system will save the student's enrollment output in a file with transcript before and after enrollment.
9- The system will output general statistics about registration problems for the department on the output file.

Alternatives

7.a - Selected course's quota is full therefore student cannot take that course. System/advisor won't approve that course, informs the student, and drops that course from student's course list.

7.b - Student haven't passed pre-requisite(s) of selected course. System/advisor won't approve that course, informs the student, and drops that course from student's course list.

      7.b.1 - In this case, the system checks whether the student can take the pre-requisite(s) course.

Object Domain Model